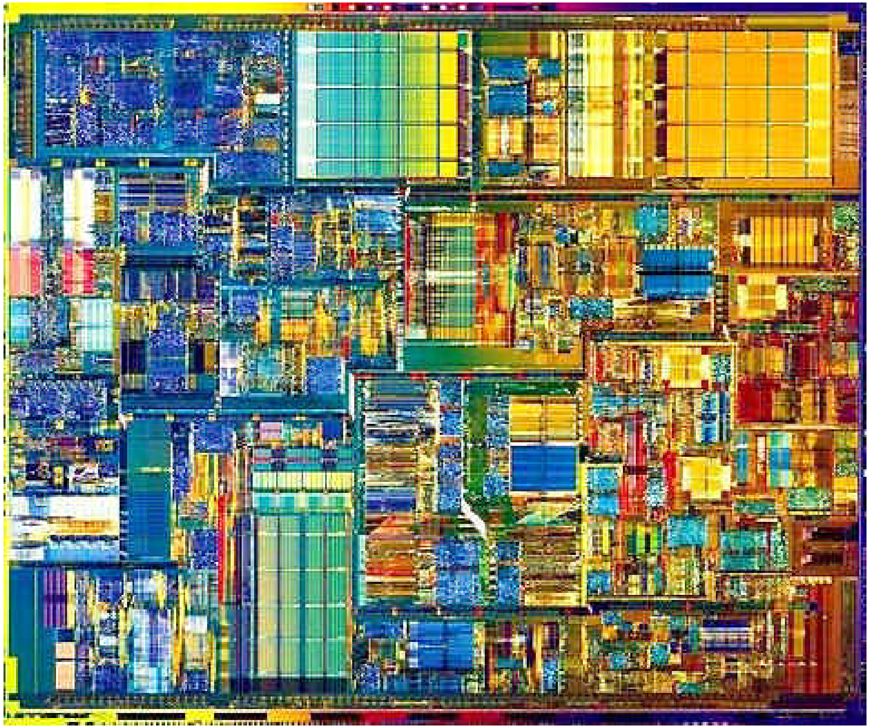
# Low Power
# Design Challenges



**Luca Mazzoni**
**ACCENT – DICO**
**Apr 04**

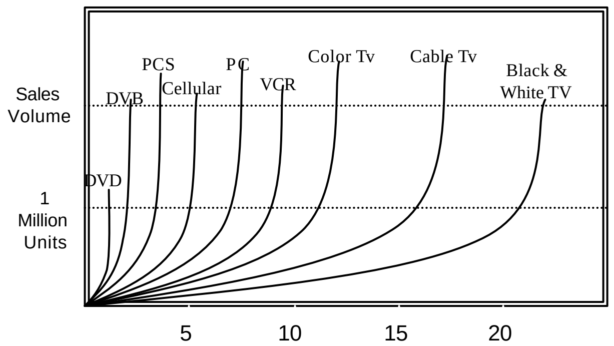
your Ideas made real

# Introduction

# Change Factors

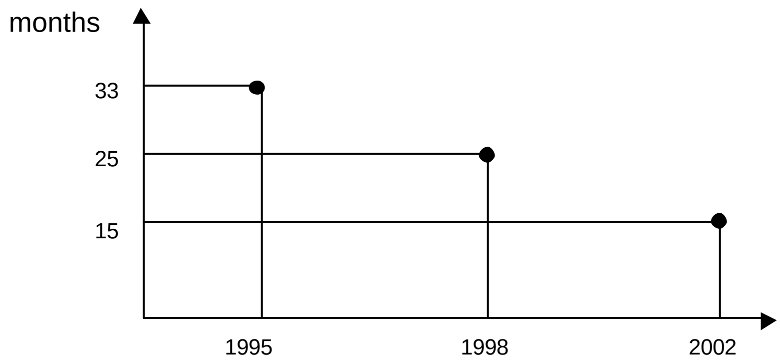
## Time to Volume

Years to 1 Million Sales




## Shrinking life time

Mobile phones



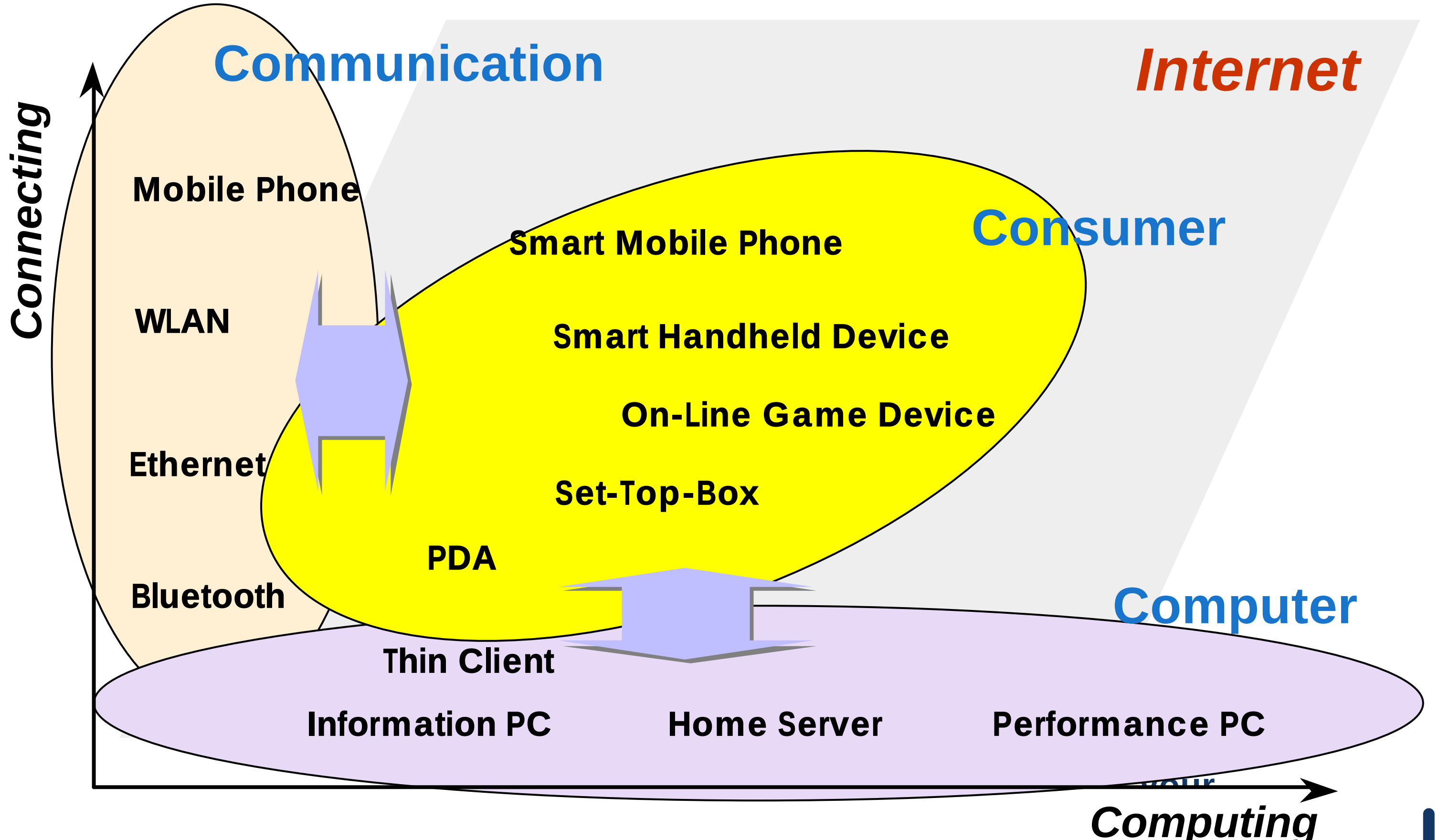

## Convergence of terminals

## Consumerization of business

- Continuous upgrade/modification of products
- Brand recognition
- Distribution channels
- Increased margin pressure
- Unpredictable market cycles

your Ideas made real

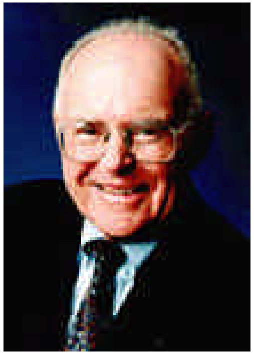
# The Driver for New Opportunities



## Technology Directions

- **In 1965, Gordon Moore predicted that transistors would continue to shrink, allowing:**
  - Doubled transistor density every 18-24 months
  - Doubled performance every 18-24 months
- **History has proven Moore right**
  - for the past 30 years Moore's Law has been a consistent indicator of semiconductor product trends
- **But, is the end in sight?**
  - Physical limitations
  - Economic limitations

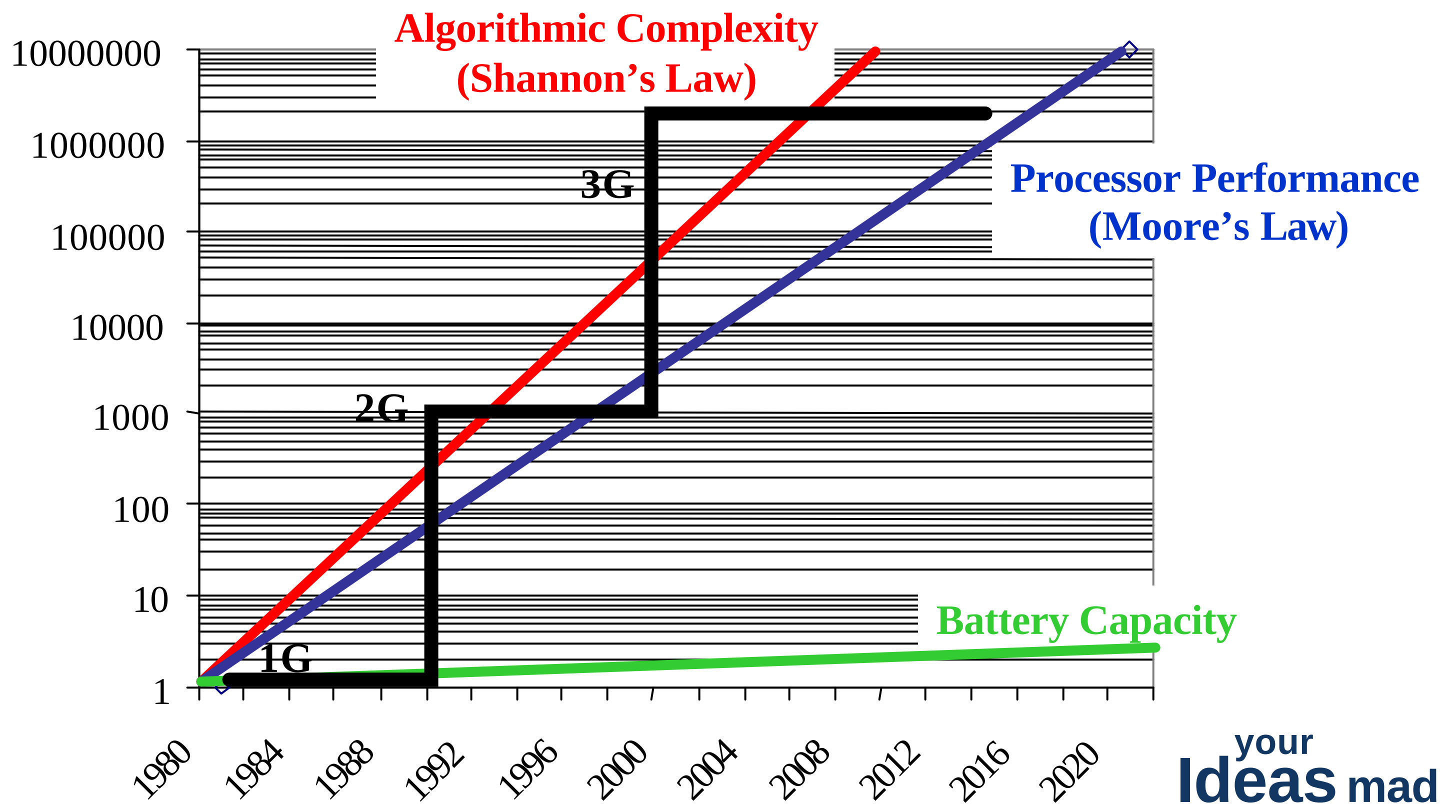
I'm smiling because I was right!



Gordon Moore
Intel Co-Founder and Chairmain Emeritus
Image source: Intel Corporation www. intel.com

# The Algorithmic Driving Force

**Shannon asks for more than Moore can deliver...**

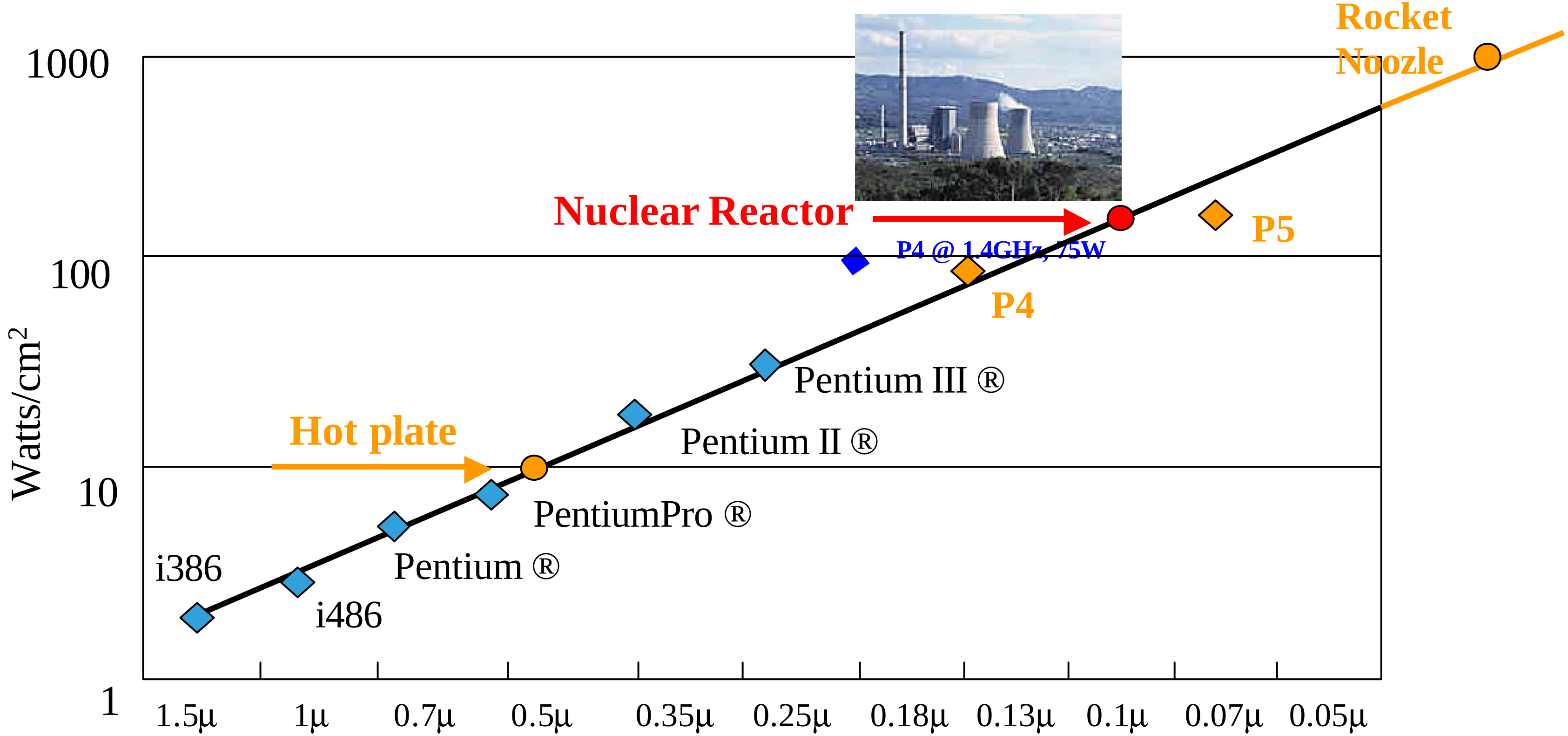


# Technology Directions: SIA Roadmap

| Year | 1999 | 2002 | 2005 | 2008 | 2011 | 2014 |
|---|---|---|---|---|---|---|
| Feature size (nm) | 180 | 130 | 100 | 70 | 50 | 35 |
| Logic trans/cm$^2$ | 6.2M | 18M | 39M | 84M | 180M | 390M |
| Cost/trans (mc) | 1.735 | .580 | .255 | .110 | .049 | .022 |
| #pads/chip | 1867 | 2553 | 3492 | 4776 | 6532 | 8935 |
| Clock (MHz) | 1250 | 2100 | 3500 | 6000 | 10000 | 16900 |
| Chip size (mm$^2$) | 340 | 430 | 520 | 620 | 750 | 900 |
| Wiring levels | 6-7 | 7 | 7-8 | 8-9 | 9 | 10 |
| Power supply (V) | 1.8 | 1.5 | 1.2 | 0.9 | 0.6 | 0.5 |
| High-perf pow (W) | 90 | 130 | 160 | 170 | 175 | 183 |

Industry Association (SIA) annually publishes and maintains its International Technology Roadmap for Semiconductors (ITRS).

## The Power Challenge



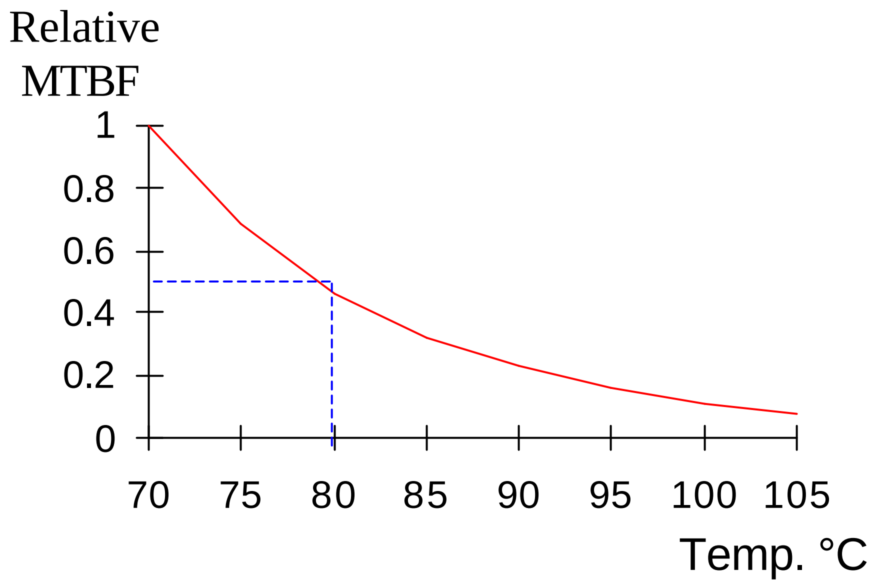

Fred Pollack, Intel
*Keynote speech*, MICRO-32

your Ideas made real

# Motivation – Power and Reliability

- **Hard Failures**
  - $T_j = Ta + (P * Q_{ja})$
  - $MTBF = f(T_j)$

- **Soft Failures**
  - Power Bus IR drop
  - Power Bus di/dit



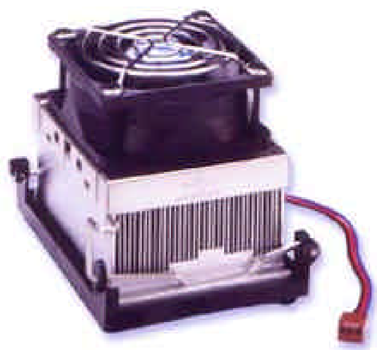

**The increment of Junction temperature by 10°C implies the 50% MTBF decrement**

**source: Intel** ***Components Quality and Reliability Handbook***

# Motivation – Save Power, Save Money

ACCENT
DESIGN TECHNOLOGY SERVICES

**CPU cooler**



**$ 22,36**

**Geforce4 cooler**

**$ 13,00**

**Passive cooler**

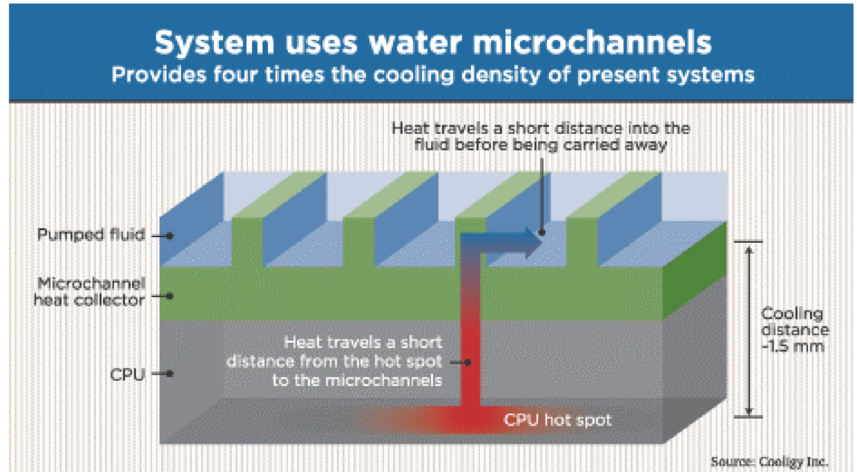
**$ 6,24**

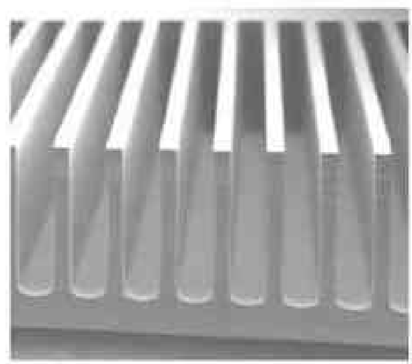

**$ 1,10**

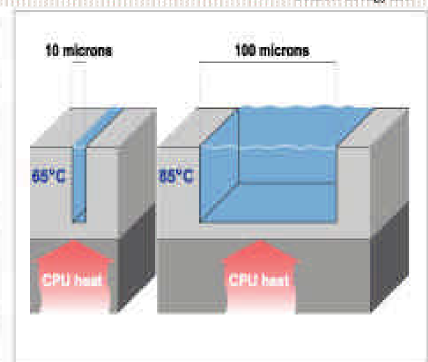
your Ideas made real

# The Cooligy system

- At 90 and 65 nanometers, the hot spots get too small for passive cooling technologies
- The fastest chips can go as high as 75 degrees C to 95 degrees C
- It can handle 1,000 watts per square centimeter; today's best passive cooling systems are specified for 250 W/cm2,
- Uses an electrokinetic pump to circulate water from the chip to the radiator. A silent and reliable solution



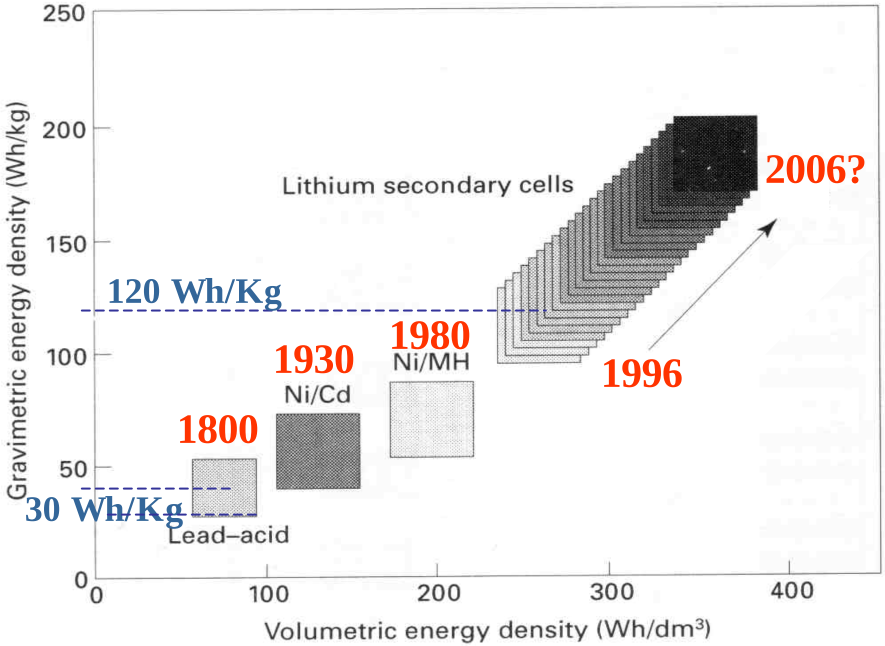



**source:** www.cooligy.com

your Ideas made real

# Motivation – Battery Power Trend

Comparison of gravimetric and volumetric energy density of lithium secondary cells with aqueous electrolyte-based systems


Battery technology has improved very slowly, and only a 20% improvement in capacity is expected over the next 10 years.

your Ideas made real

# Summary of Motivations

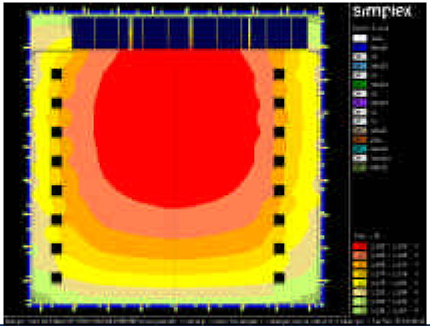
- Deep sub-micron technology
  - smaller geometries $\Rightarrow$ higher device densities, clock frequencies
  - high density chips: greater power consumption in spite of lower voltage supply
  - reliability issues (IR drop, heat dissipation, electromigration ..)
  - electromagnetic compatibility (RF shielding and de-coupling components)

- High-power consumption
  - rack-mounted computing systems, commercial servers (3kW)

- Portable applications
  - Mobile User paradigm, *"Optimally Connected Anywhere, Anytime"*
  - constraint is battery life-time (10-15% increase per year)
  - Second order effect is that higher current draws decrease effective battery energy capacity

# Summary of Motivations

- Innovative applications
  - Micromechanics, smart sensors, bioinformatics, genetics, healthcare…
- Financial
  - Reducing packaging, PCB, power supply, heat-sink costs (high volumes)
  - Achieving energy savings
- Reliability
  - Power dissipated as heat, higher temps reduce speed and reliability
  - Current draw causes IR drops in power supply voltage
  - Switching current (dI/dT) causes inductive power supply voltage bounce µ LdI/dT

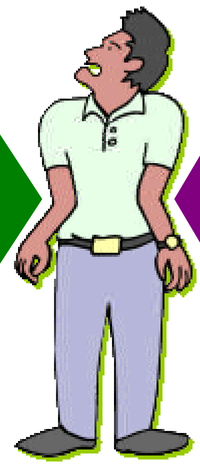



your Ideas made real

# Constant Design Squeeze

## Product Requirements

- Shorter Time to Market
- Shorter Product Shelf Life
- Product Complexity Increasing
- Feature Competition



*Designers*

## Technical Constraints

- SoC Integration
- Design Complexity
- Power & Speed Requirement
- Packaging & Testing
- Die Size Economy

your Ideas made real

# Low Power
# Design Challenge

# Low power design challenge

**The challenge**

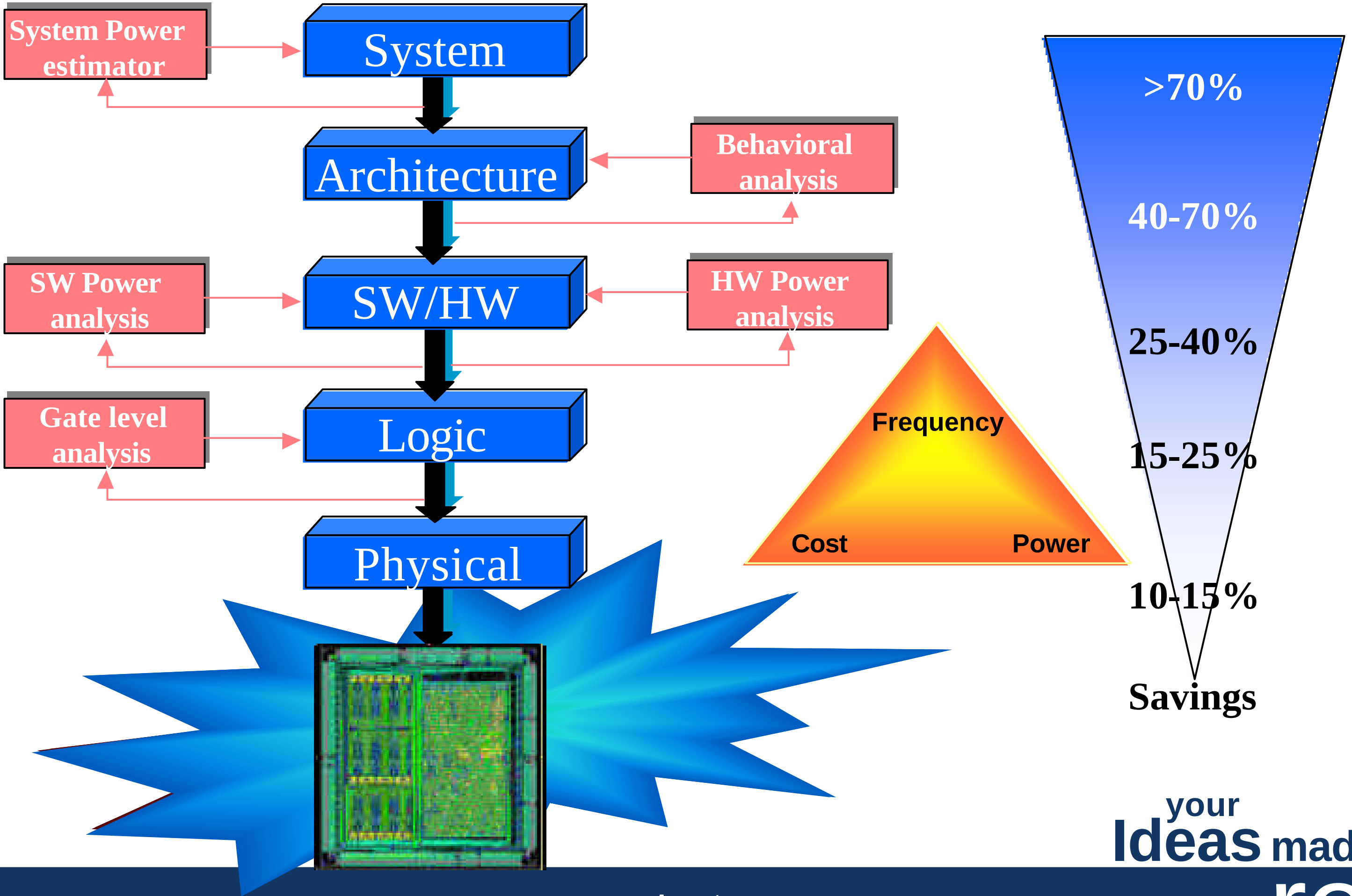
**"To design an electronic system (HW *and* SW) that provides the target functionality with minimum power consumption"**

**The solution**

**From the system concept down to the implementation phase, adopt a design style that includes power consumption as a figure of merit, and exploit all the opportunities and techniques available at each design level to reduce it**

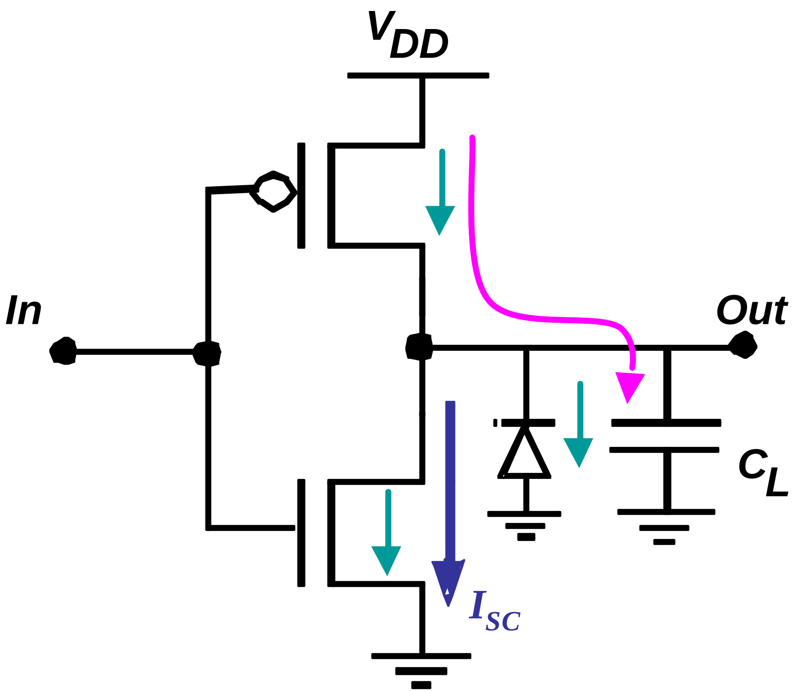
your Ideas made real

# Power saving opportunities




your Ideas made real

# Power Ingredients




- **Dynamic Dissipation**

  $P_{dyn} = a C_L V_{DD} F$

- **Short-Circuit Currents**

  $P_{sc} = V_{DD} I_{sc}$

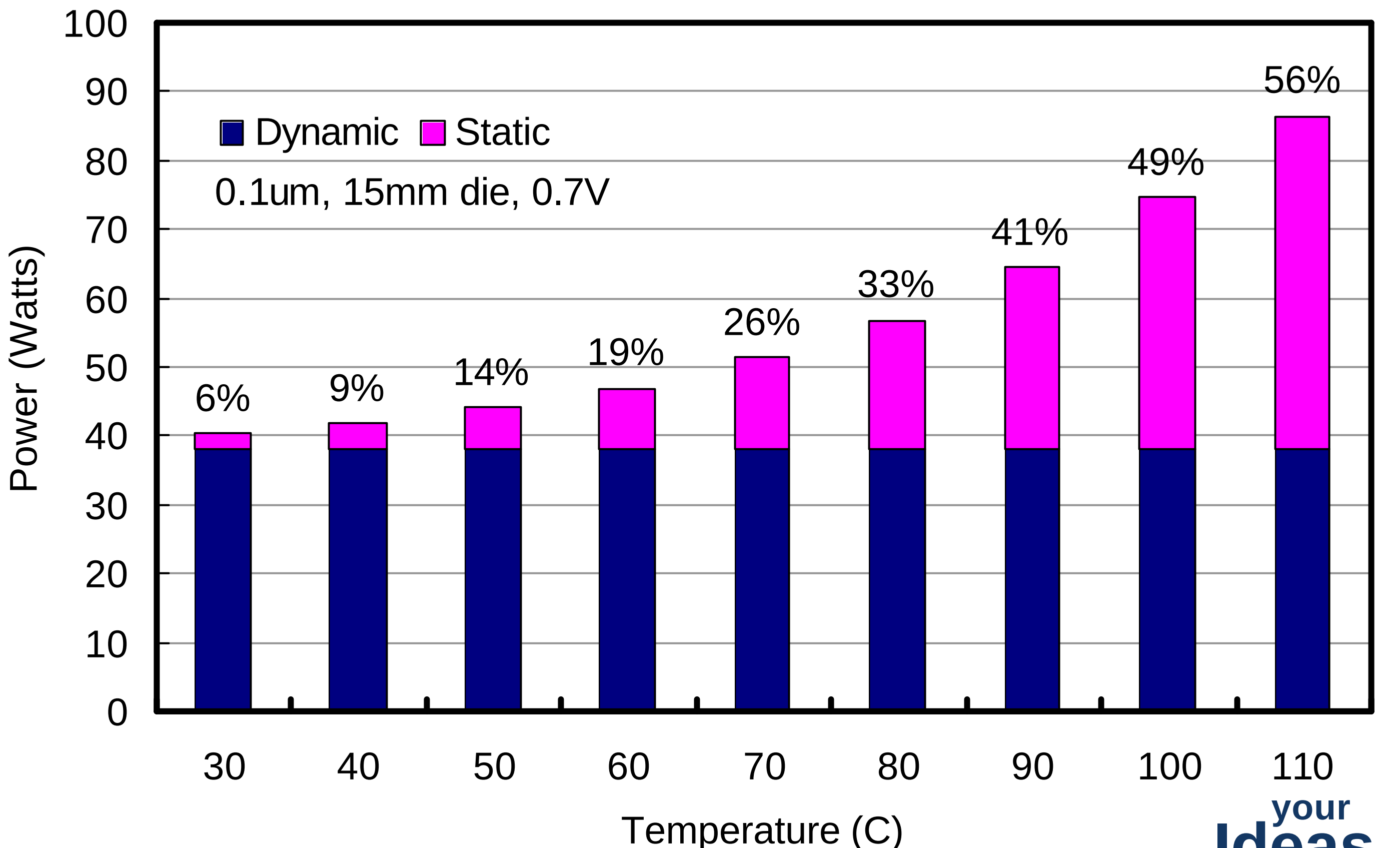
- **Static Dissipation**

  $P_{stat} = V_{DD} I_{leak}$

**Coming soon (90 and 65 nm):**

- ❑ **Gate leakage, a tunnelling current through gate oxide insulation**

# Trends of Power Dissipation in Temperature



# System / Application level optimization

# Application Level Optimization

- **Analyze the target application**
  - how the user exercises the device
  - how the environment exercises the device

- **Detect functional states; profile their power consumption; modify/introduce new execution modes that consume less power**
  - find out opportunities for working at lower *energy* cost
  - E.g. turn off the display's backlight of a portable music player
  - ✓ E.g. use lower precision/resolution
  - ✓ E.g. introduce dynamic power managers

    (turn the monitor off, stop/slow down hard disk spinning etc)

> • **Display no/small images**
> • **MP3 full/lossy decompression**

- **This approach brings biggest savings, but solutions are specific**
  - customize general indications for the case in hand

# Customer case study

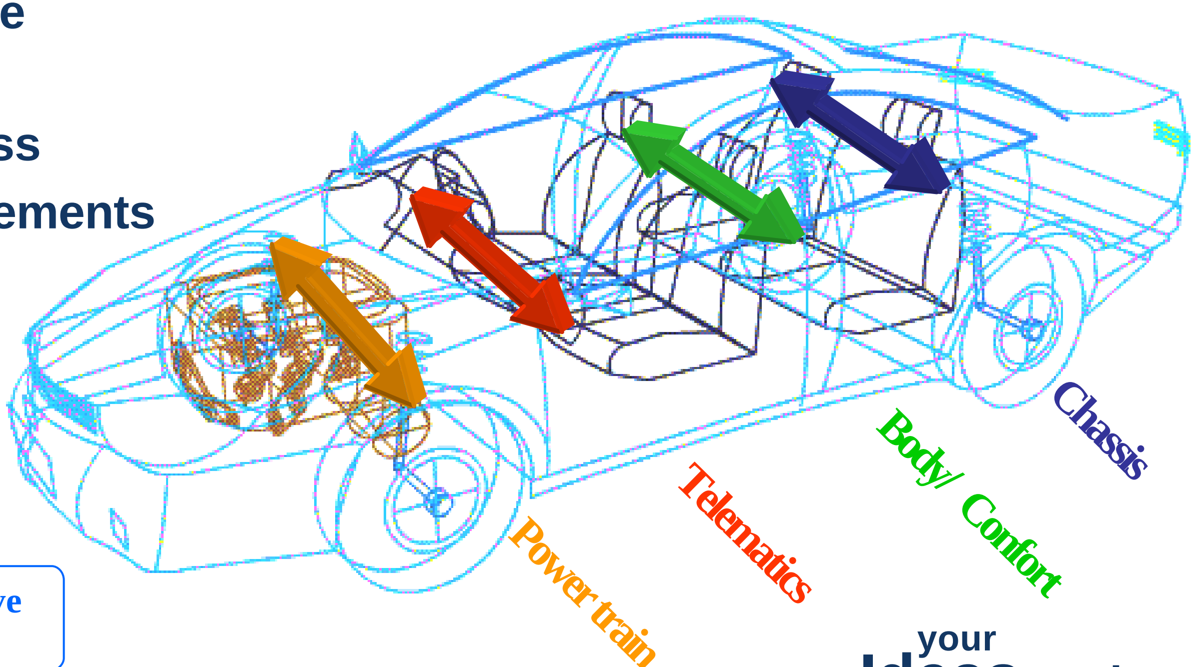
## Project objective:

add an *efficient* and *unified power management strategy* on each electronic controller unit to reduce power consumption in stand-by mode

.. but also address additional requirements

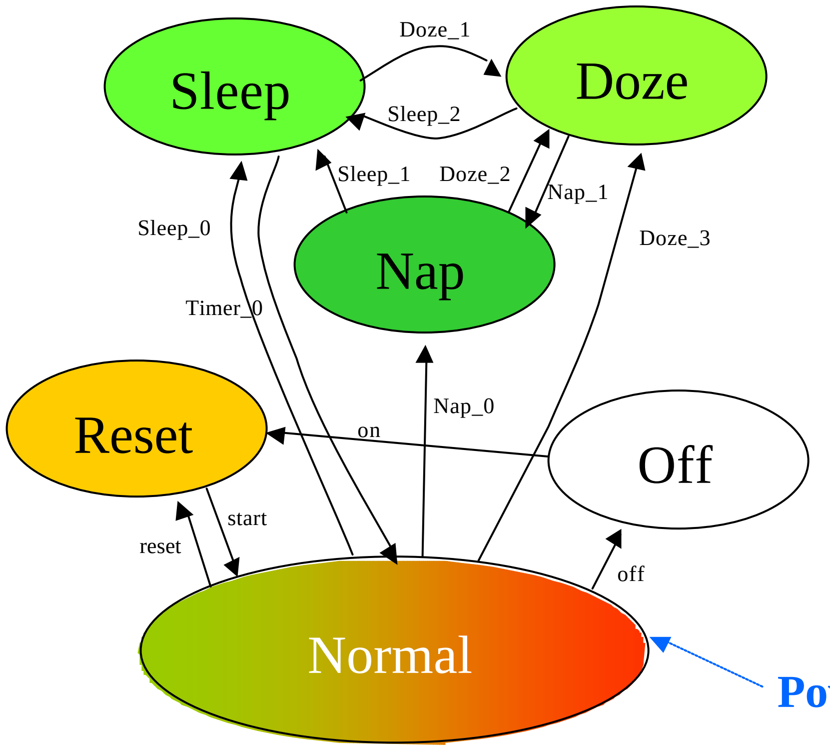
+ Flexibility
+ Cost
+ Safety

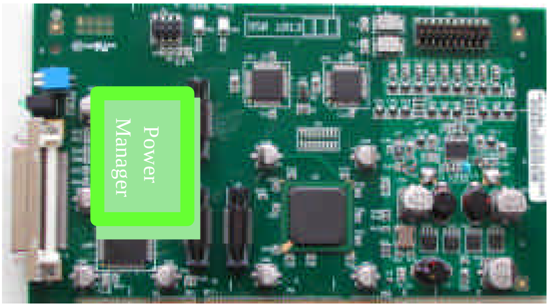
Always a multi-objective problem




your Ideas made real

# System Solution




Power State Machine

# Architecture level

## Dynamic
## Power & Energy
## Management

**An electronic system is a combination of:**

- Hardware:
  - Computation/storage/communication units, peripherals
- Software :
  - Application code and system (O.S., stacks, HAL) code

- **Energy is required by all hardware units**
- **Software organization affects how hardware consumes energy**

Opportunities for power optimization

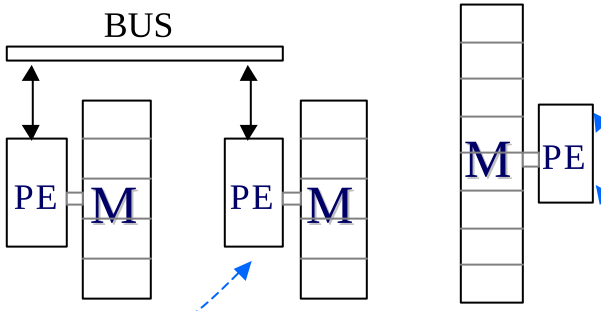
- **System Dynamic Power Management controls all the units**
  - O.S., HW power manager

your Ideas made real

# Architecture Level Design

- **HW/SW codesign: allocate processing elements**
  - Choose the most power efficient IPs
    - RISC, DSP, Configurable Processors, Reconfigurable Coprocessors
    - Dedicated processing engines (multimedia codec, baseband modules)
    - Memories, memory interfaces, I/O interfaces…
- **Explore the bindings between functional tasks and processing elements; try scheduling the tasks**
  - structural approach (platform extension and optimization)

**e.g. TCP/IP protocol stack: 16 mW @ 100MB/s**




**MPEG4 decoder: 2-6 mW QCIF @ 15 f/s**

**e.g. 32 bit RISC [mW/MHz], 0.13 um:**
**0.05 (core) <=> 0.4 (core, caches, MMU)**

**e.g. same 32 bit low power RISC [mW/MHz]**
1) 0.18 um Tech.: 0.240
2) High Speed version, 0.295 (+22%)
3) Other Tech.: 0.055 ( -77% )

# Dynamic Power Management (1/2)

- **Circuits and systems are:**
  - Designed to deliver peak performance, but …
    do not need peak performance most of the time
- **Dynamic Power Management (DPM):**
  - Shut-down idle components
    - clocks off / supply off
- **Dynamic Energy Management (DEM):**
  - Reduces Energy/Charge consumption by dynamically adapting performance levels to the current workload

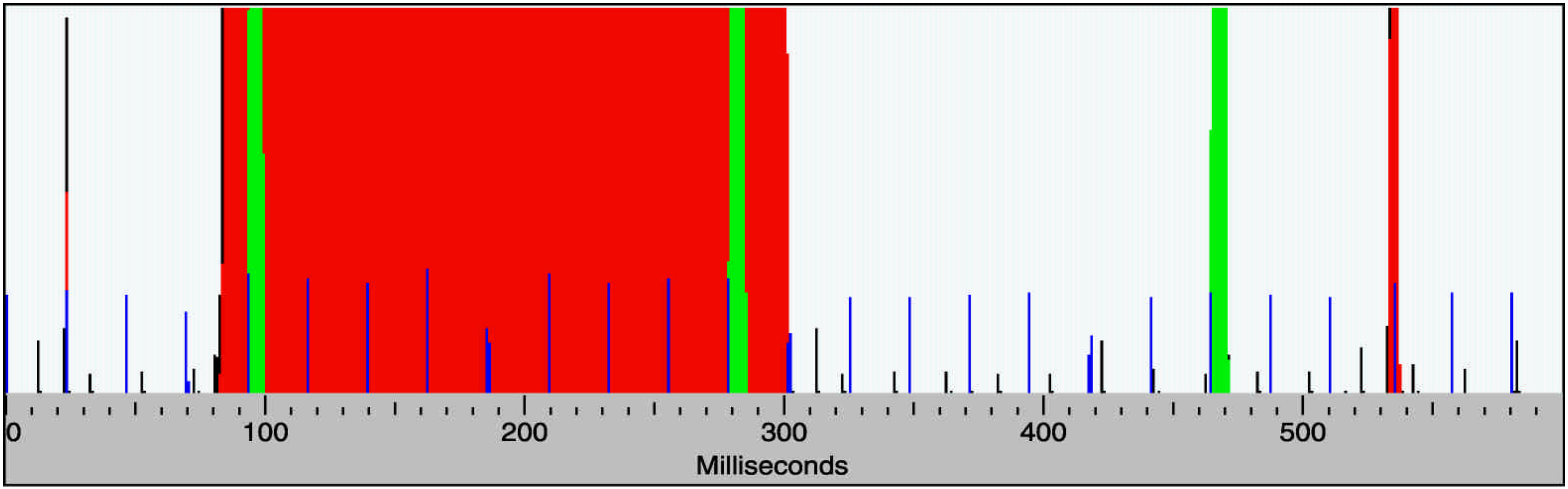
    ⇒ Dynamic Voltage Scaling (DVS):
    - slow-down components - scale frequency *and voltage*

Dynamic Power has a $x^2$ dependency on supply voltage

- **Problem:**
  - Analyze how the system is managed dynamically
  - Power control policies (predictive, adaptive…)

your Ideas made real

# Episode classification




Interactive (Acrobat Reader), Producer (MP3 playback), and Consumer (sound daemon) episodes.

your Ideas made real

# Dynamic Power Management (2/2)

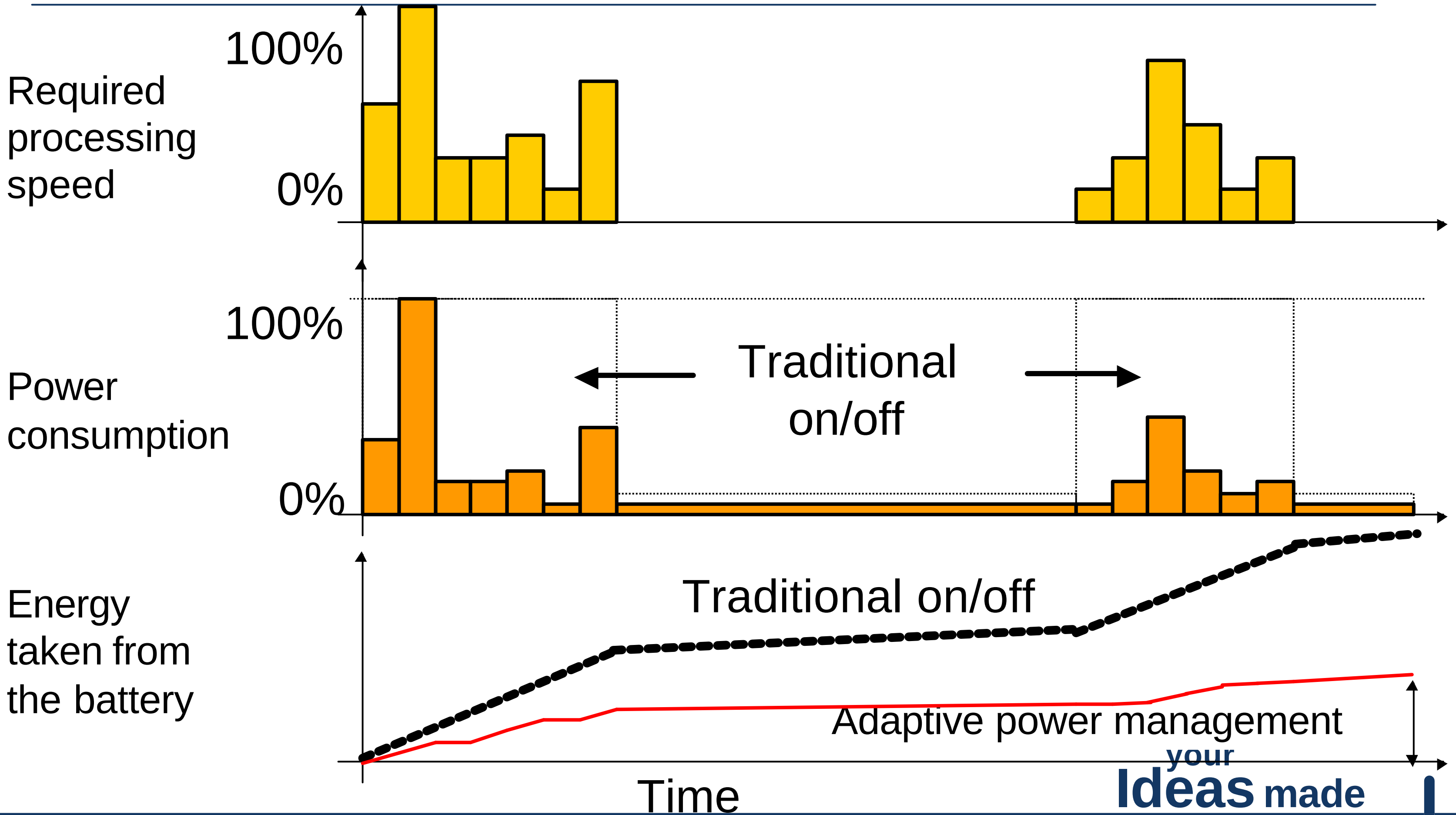
- **In devices with an Embedded Operating System, the OS knows which tasks are running or waiting. It can therefore, perform the DPM decisions…**

  **… but HW should provide full control to SW**

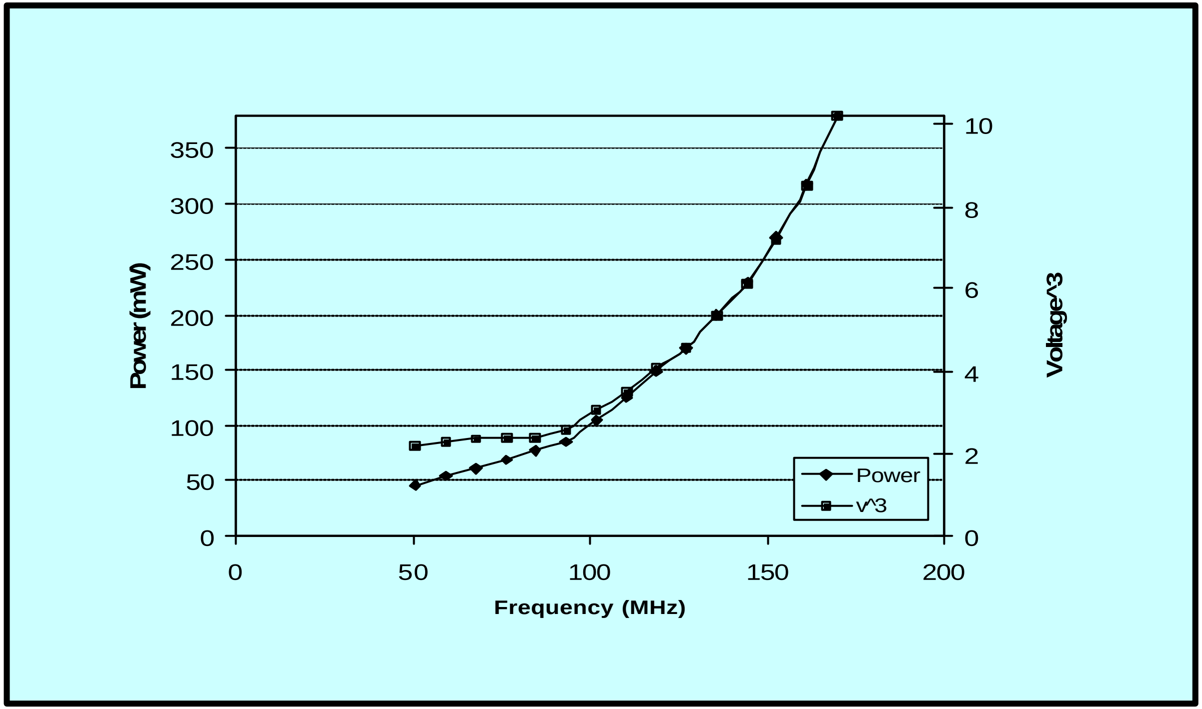
  - Memory mapped activity monitors /exception triggers
  - Memory mapped clock gating enable/disable and frequency scaling
  - Wake-up module with programmable sources
  - Snooping logic (wake-up conditions)
  - Level shifters at the voltage domain boundaries
  - Control interface to drive external Voltage Regulator

- **Dynamic Battery Management**
  - Maximize battery elapsed and total life-time by applying DPM closed loop control based on battery's behaviour (residual charge, time to empty,etc.)

# Adaptive Power Management



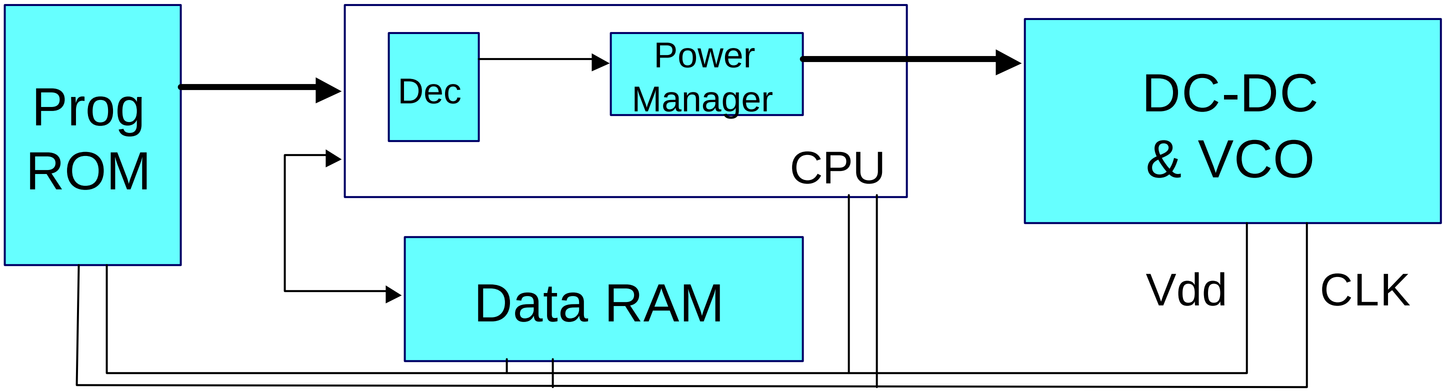
# Voltage and F scaling save energy




- Just-in-time computation
  - Stretch execution time up to the max tolerable

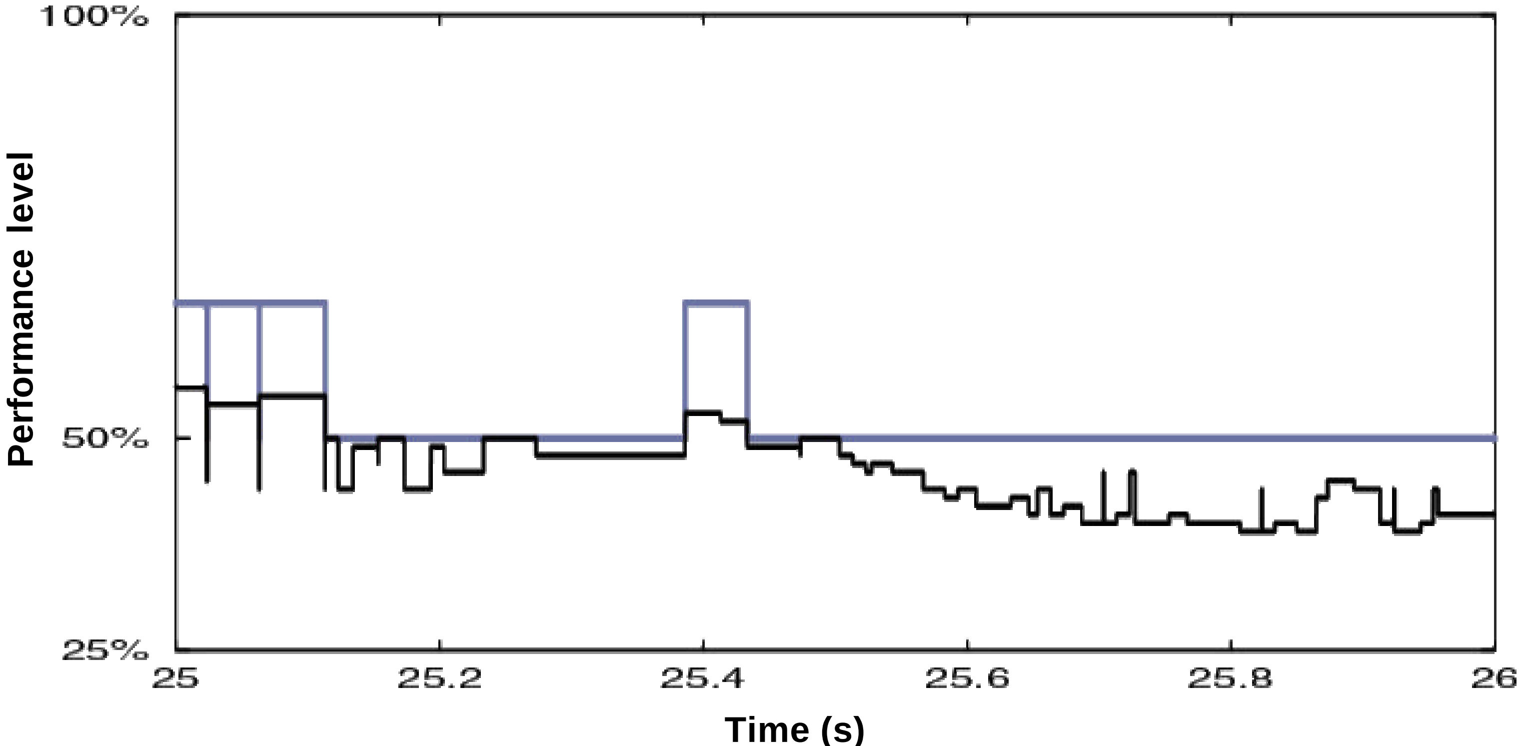
# Variable-supply Architectures

- **High-efficiency adjustable DC-DC converter**
- **Adjustable synchronization**
  - Variable-frequency clock generator [Chandrakasan96]
  - Self-timed circuits [Nielsen94]
- **Example: Power-pro architecture, Crusoe embedded processor**




your Ideas made real

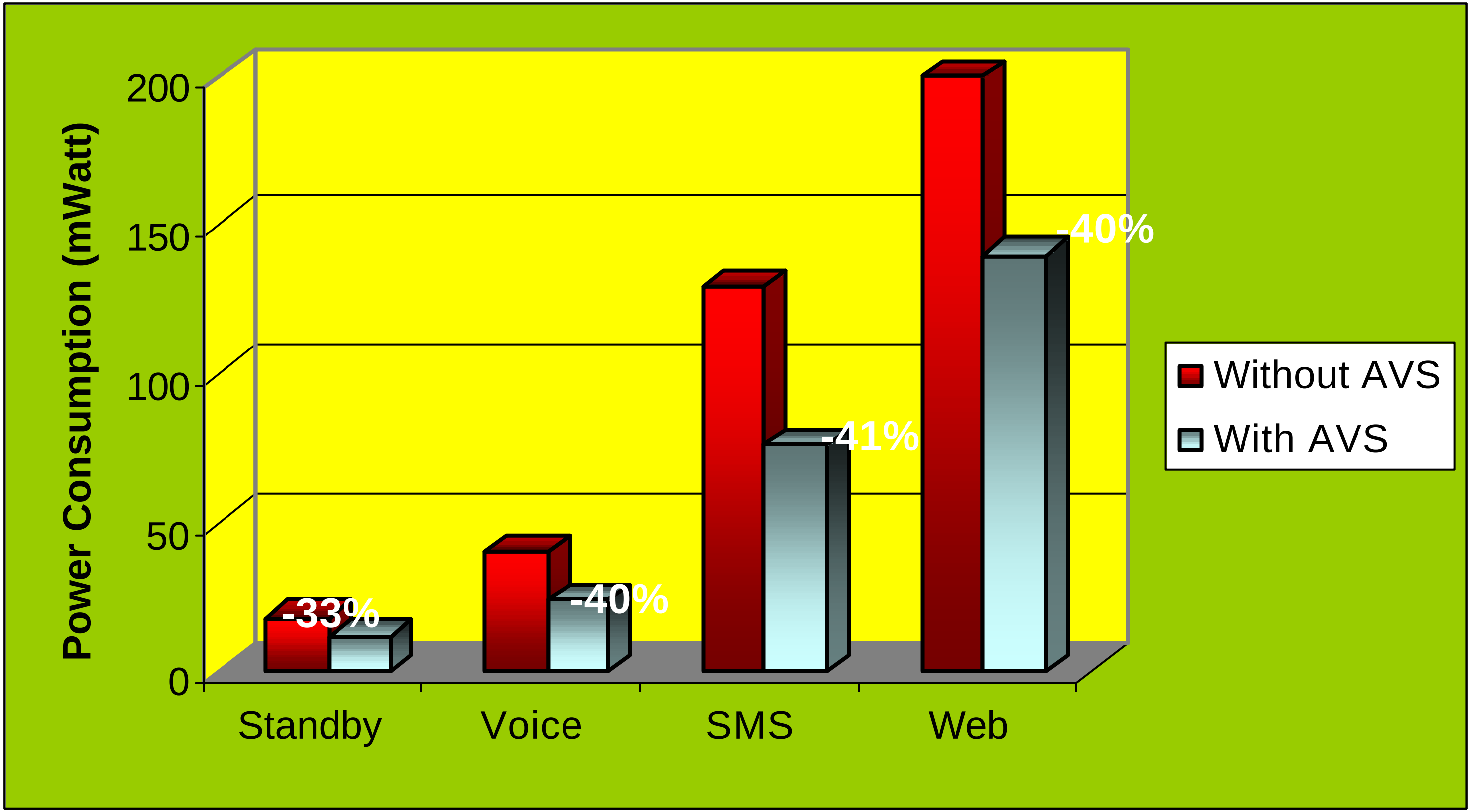
# Plaympeg: Red’s Nightmare (segment)




**source: ARM** Intelligent Energy Management

your Ideas made real

# Power Savings in a Baseband




**source: ARM** Intelligent Energy Management

your Ideas made real

# Intel Mobile Pentium

Table 1: Maximum Performance vs. Battery Optimized Mode

| Maximum Performance Mode | | | Battery Optimized Mode | | |
|---|---|---|---|---|---|
| Frequency | Voltage | Max. Power Consumption | Frequency | Voltage | Max. Power Consumption |
| 500 MHz** | 1.1 V | 8.1 W | 300 MHz | .975 V | 4.5 W |
| 600 MHz** | 1.1 V | 9.7 W | 300 MHz | .975 V | 4.5 W |
| 600 MHz* | 1.35 V | 14.4 W | 500 MHz | 1.10 V | 8.1 W |
| 600 MHz | 1.6 V | 20.0 W | 500 MHz | 1.35 V | 12.2 W |
| 650 MHz | 1.6 V | 21.5 W | 500 MHz | 1.35 V | 12.2 W |
| 700 MHz | 1.6 V | 23.0 W | 550 MHz | 1.35 V | 13.2 W |
| 750 MHz | 1.35 V | 17.2 W | 500 MHz | 1.10 V | 8.1 W |
| 750 MHz | 1.6 V | 24.6 W | 550 MHz | 1.35 V | 13.2 W |
| 800 MHz | 1.6 V | 25.9 W | 650 MHz | 1.35 V | 15.1 W |
| 850 MHz | 1.6 V | 27.5 W | 700 MHz | 1.35 V | 16.1 W |
| 900 MHz | 1.7 V | 30.7 W | 700 MHz | 1.35 V | 16.1 W |
| 1 GHz | 1.7 V | 34.0 W | 700 MHz | 1.35 V | 16.1 W |

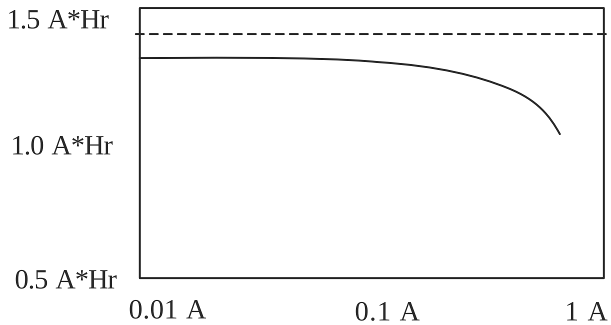
**source: Intel**

# A glance to batteries

- **While waiting for reliable Power Scavenging techniques…**

  **Let's model the Battery :**

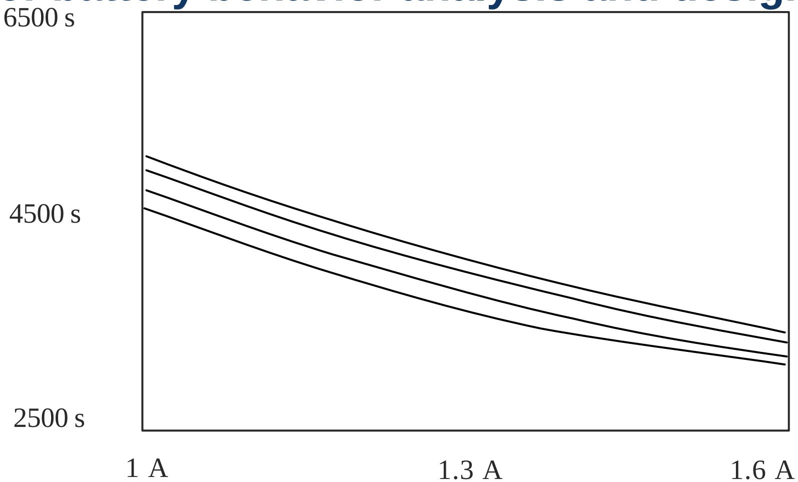
  - traditional techniques for low power design assume a constant voltage supply that delivers a fixed amount of energy
  - not valid for battery-operated devices
  - the higher the average discharge current, the higher the energy waste of the battery (see constant current load)

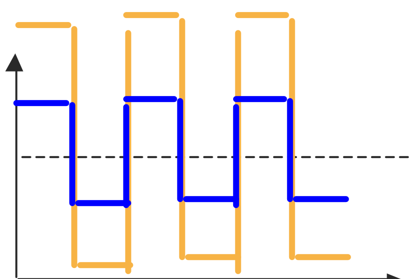



your Ideas made real

# A glance to batteries

- **Actual battery life-time depends also on the profile (over time) of the current load**
  - A smaller average current can discharge the battery faster than a larger one, if current levels are more skewed from the average value
- **Battery models: discrete time model (VHDL) for Lithium-Ion batteries:**
  - first order effect (state of charge, discharge rate and frequency)
  - second order effect (output resistance, temperature)
- **useful for battery behavior analysis and design exploration !**



your Ideas made real

# Battery Driven Power Management

- **Monitor the State of Charge (SOC) of the battery and decide when reduced power states are entered**
- **it can be combined with work-load driven policies**

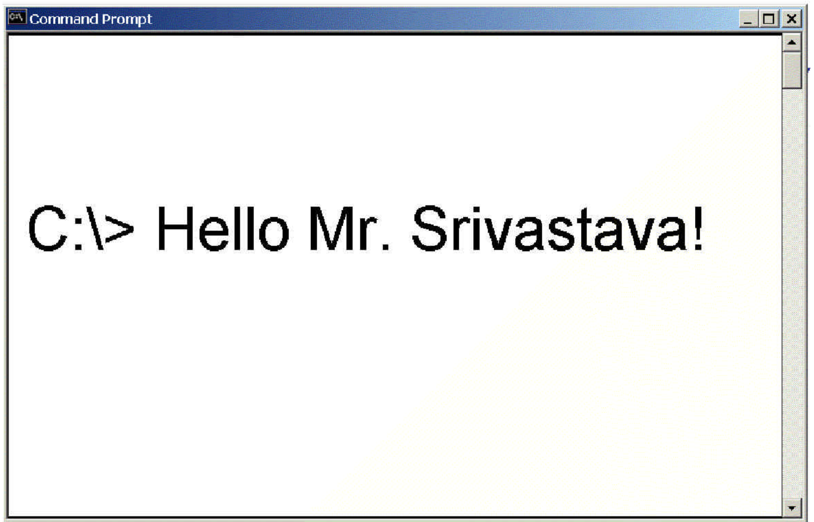
- **Example: MP3 Digital Audio Player Battery Driven DPM**
  - SOC above 50% (i.e. Vdd > 3.7 V)
    Full decompression (44KHz samples/s, 128 kbps, 87mW
  - SOC between 50% and 20% (i.e. Vdd > 3.3 V)
    Lossy decompression (22.05KHz samples/s, 64 kbps, 50mW
  - SOC below 20% (i.e. Vdd < 3.3 V)
    sleep (1 mW)

| | No WDDPM | WDDPM |
|---|---|---|
| No BDDPM | 66683 | 89570 |
| BDDPM | 74340 | 100457 |

# Low Power Design Techniques
# SW

# The crucial question

- **How much is the power?**




$$P = a \times CV^2 \qquad P = I \times V \qquad \mathbf{P = ?}$$

your Ideas made real

# The design flow

ACCENT
DESIGN TECHNOLOGY SERVICES

**Target application**

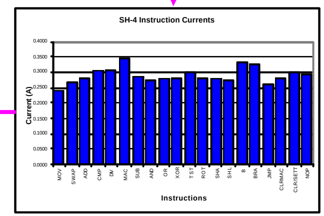
C:\> Hello Mr. Srivastava!

**SW libraries and Coding style strategy**

```
#include <stdio.h>
Void main (void)
{ printf("Hello Mr. Srivastava!");}
```

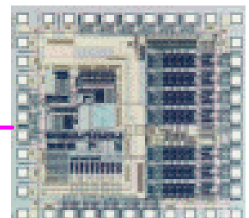
**Target processor policy and compiler optimization strategy**

```
MOV    ip,sp
STMDB  sp!,{fp,ip,lr,pc}
SUB    fp,ip,#4
CMP    sp,sl
BLMI   __rt_stkovf_split_small
ADD    a1,pc,#L000024-.-8
BL     _printf
MOV    a1,#0
LDMDB  fp,{fp,sp,pc}
```

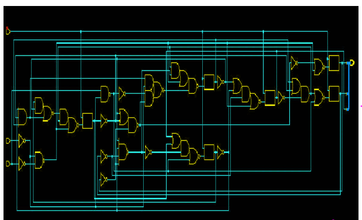




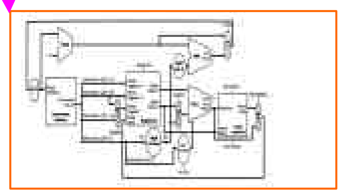
**Power analisys and task allocation strategy**



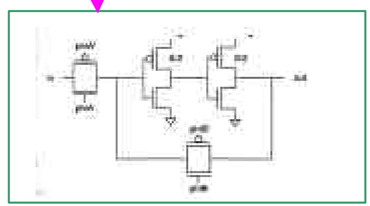
**Silicon Implementation strategy**



**Circuitry design and Synthesis strategy**



**Architecture strategy**



**Cells library**

your Ideas made real

# Behavioral Level Design

**To realize the target functionality, select the most energy-efficient way to implement the processing algorithm, among those ways that respect area/timing constraints**

- **Different Memory usage**
  - Trade off processing speed and memory area
- **SW**
  - Data type, data flow, flow control
    - Array, pointer array, binary array, linked list
  - SW writing style and techniques (see later)
- **HW**
  - Datapath plus controller unit
  - Many degrees of freedom: parallelism, pipeline stages, computation precision, etc

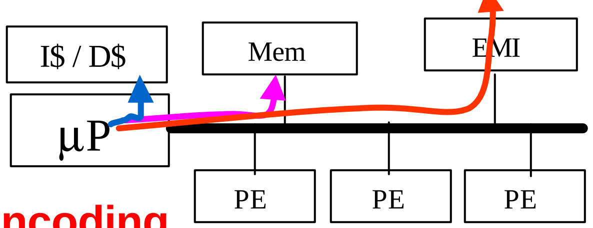
your Ideas made real

# Embedded SW Coding

- **Software coding guidelines: minimize power due to data transfer and data processing**
  - SW data organization (e.g. stack, heap, data placement)
  - Reinforce spatial and temporal locality (e.g. caches)
  - Loop transformations
  - Abstract data type to minimize memory power
  - Coding style
  - Exploit ISA specific instructions
  - Compiling techniques
  - Code density optimization
  - *Exploit underlying system architecture*

More **specialized** code:
✓ **less** portable
✓ more **effort**
✓ **better** results

$E_{L0}$ = 1.5 nJ    $E_{L2}$ = 7.0 nJ
$E_{L1}$ = 3.0 nJ    $E_{RAM}$ = 127 nJ




**Memory organization and data / instructions encoding play major role due to traffic reduction**

# Software optimizations

## High level

- **Linear loop transformations**
  - Loop permutation
  - Loop skewing
  - Loop reversal
  - Loop scaling
- **Iteration space tiling**
- **Multi-loop transformations**
  - Loop Fusion/Fission

## Low level

- **Instruction selection**
- **Reorder instructions**
  - Reduce inter-instruction effects
  - to minimize Hamming distance
- **Register assignment**
  - Minimize spills to memory

- **Compiler options: fast code** *usually* **consumes less**

**Ex: MPEG4 decoder**

- **Source code optimizations give as much as 32.3% energy savings**
- **Compiler optimizations account for only 0.6% energy savings**

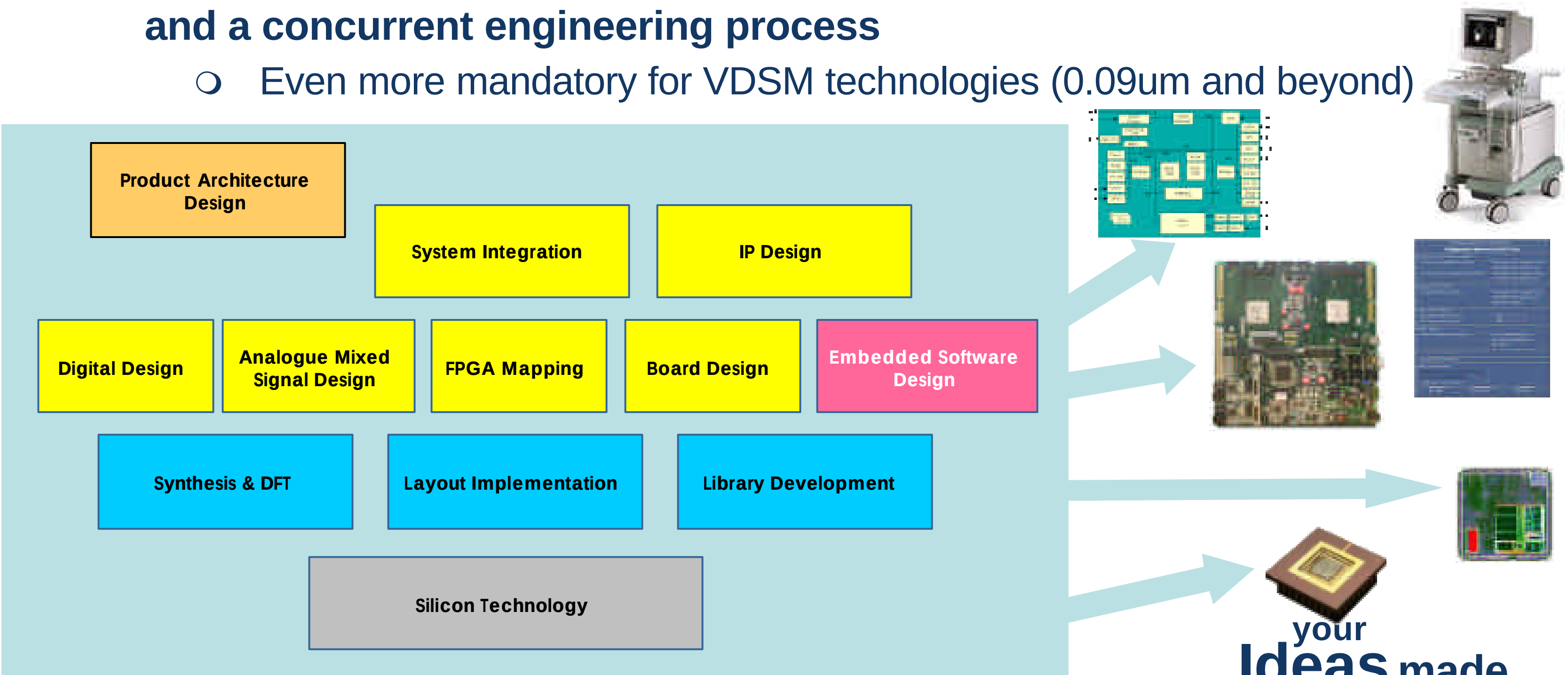
your Ideas made real

# Low Power
# Design Challenges
# Conclusions

your
Ideas made
real

# Conclusion: *think* low power

- **Low power design methodology is a way to approach the system specification and design, a coherent design style.**
- **It requires cross-area competences, a global vision of HW/SW, and a concurrent engineering process**
  - Even more mandatory for VDSM technologies (0.09um and beyond)




your Ideas made real

# Conclusion: *design* low power

- **Applying ideas**
  - Identify opportunities as early as possible in the design flow
  - Design the SW components/mechanisms to exploit them
  - Design the HW components/mechanisms to exploit them

- **Complete and coherent** design methodology
  - Combine EDA tools in a robust design flow
  - Check that EDA tools are able to apply specific features
    - Sometimes they do not work
    - Change the design entry to let the tool work
    - Find workarounds, e.g. by driving the tool with TCL scripts
  - Continuously experiment new tools